UNITED STATES PATENT APPLICATION

OF

JUNAID ISLAM, HOMAYOUN VALIZADEH, JEFFREY S. PAYNE

FOR

PROGRAMMABLE NETWORK APPLICATION SERVER

PREPARED BY WILSON SONSINI GOODRICH & ROSATI

BACKGROUND OF THE INVENTION

Field of the Invention

This invention relates to the field of networking. In particular, the invention relates to programmable devices used to implement applications and services on a network.

Description of the Related Art

The Inadequacies of Pre-Programmed Network Devices

Existing network environments are characterized by a disjunction between programmable components, which are generally CPUs in workstations connected to the network, and pre-programmed units in the infrastructure of the network, such as routers and switches. By design, these pre-programmed network devices are closed from the perspective of network users and service providers.

The rigidity of pre-programmed network devices results in inefficiencies in the maintenance of networks and inflexibility in the deployment of new services or enhancement of existing services. For instance, the provisioning of new applications at a node in a network typically entails the overhead of one or more of the following: 1) developing hardware to support the new applications 2) writing new software for existing network platforms to support the desired applications 3) deploying workforce to the network node to install hardware and/or software developed to support the desired applications 4) interrupting or re-routing traffic that would otherwise pass through the device while the device is upgraded with the new hardware and/or software.

The prior art does include some network devices in which parameters may be changed via a network, without requiring the network device to be restarted or interrupting traffic through the device. One such example is Cisco IOS™. Such systems, however, only allow parameters to be adjusted without restarting the device. They do not allow for the addition or deletion of software modules without interruption to network services.

As a result of this inflexibility, network service providers are constrained in the geographical breadth of their services by physical resources. As personnel must be dispatched to install and administer existing network devices, service providers are constrained to offer services only where they have sufficient manpower and physical resources. Consequently, there are currently no network service providers with global reach.

The Coupling of Hardware and Software in Existing Network Devices

The pre-programmed nature of existing network devices also results in a tight coupling between hardware and software used on the network devices. New application modules may not be dynamically added to the vast majority of network devices, as such devices typically utilize a single, monolithic program which executes a finite set of services. Though routers have been developed for platforms such as Windows NT ™, such technologies are too slow for widespread use in service provider networks and do not allow for the dynamic loading and unloading of applications without interrupting packet forwarding. As such, to provide new services, service providers are often forced to replace existing network devices with new devices that include software for the

respective service, a process that may take years. The replacement of boxes to support new functions has grown particularly problematic, as the period for amortization of network devices is shrinking. As such, the coupling of hardware and software results in an onerous financial constraint on service providers.

Moreover, the coupling of hardware and software on network devices precludes third parties from developing applications for the devices. Given existing network technology, third parties wishing to develop new applications for the devices would have to co-operate with the device manufacturers to have their software included in the device prior to deployment. Existing network devices make no provisions for the inclusion of new modules after deployment. As the development of new services accelerates, network devices become obsolete before generating an adequate return on investment.

Inability to Place Agents on Existing Network Devices

The inability to load modules, or agents, on existing network devices presents difficulties in the analysis of network parameters. Existing network devices do not allow agents to be uploaded in order to analyze or act upon network traffic. An example of this inefficiency is evident in existing support of Service Level Agreements (SLAs). Existing SLA techniques typically utilize SNMP or an another architecture which polls network devices periodically to read counters. Such data is collected and then transported over the network for post-facto analysis, i.e., to determine packet discard rate and other relevant

parameters. This architecture demands substantial overhead to scale to a large number of devices and does not offer traffic analysis in true real-time.

The inadequacies of current network devices evince a need for reprogrammable devices that support multiple network management functions. Code supporting network management functions should be dynamically loadable on network devices, thereby alleviating the need to physically install new devices at network nodes. Devices should also be configurable remotely in order to eliminate the costs of deploying manpower to service the devices. Such devices should also be scalable to accommodate network expansion, and should facilitate load balancing and redundancy.

SUMMARY OF THE INVENTION

The present invention comprises an architecture for programmable network servers. The network server is capable of executing software modules resident on its hardware to support assorted applications and network management services. These modules may be dynamically loaded, unloaded, or modified to facilitate service changes without interrupting network traffic routed through the device. Such unloading and loading of modules can be administered remotely, via a network backbone, service provider network, LAN, or other internetwork coupled to the device. Alternatively, administrators can change the operating parameters of individual management modules via the network to affect performance gains or modify existing operating requirements.

In embodiments of the invention, the network application server may reside at the edge of a service provider network and fan out to subscriber LANs. In other embodiments, the network application server may be located at a customer site and connect to the service provider network via the customer's Local Area Network. In some such embodiments, the network application server may tunnel to the service provider network via a Virtual Private Network, or VPN.

The invention allows service providers to administer network application servers and upload new modules remotely. These modules may emulate legacy systems, provide VPN services such as tunneling protocols, support network management functions, or provide new types of applications developed by network service providers or third party developers. By enabling the remote uploading of new modules, the invention helps to eliminate the lag time in the provision of new network services. Likewise, by allowing service providers to administer the network servers remotely, the invention pre-empts the necessity of allocating service provider personnel to subscriber sites.

By decoupling hardware and software on network application servers, the invention allows hardware and software components to be retailed to subscribers separately. This feature of the invention also allows third party development of modules for network services.

Embodiments of the invention employ a multi-tiered software architecture comprising a forwarding engine, an application tier, and a network management tier. In embodiments, the forwarding tier is responsible for

forwarding packets between a service provider network and a subscriber LAN coupled to the network application server. In embodiments, the forwarding engine also includes encryption and authentication mechanisms for accessing modules in the network application device. The forwarding engine is also a conduit between modules resident on the network application server and data packets traversing the network application server.

The application tier contains modules for networking applications. Such applications may correspond to VPN functions, including but not limited to applications such as Multiprotocol Label Switching, or MPLS, Layer Two Tunneling Protocol, or L2TP, and IP Sec. This allows the network application server to emulate any type of VPN. The modules may also be unrelated to VPNs, and support applications such as Traffic Shaping or Multicasting. Modules in the application tier may also be encoded to support entirely new types of applications.

A third tier in the software architecture comprises a network management tier. Modules in this tier may support remote network monitoring and management protocols, such as the Simple Network Management Protocol (SNMP) and the Common Management Information Protocol (CMIP). Modules may include support for CORBA Object Request Broker or an XML based messaging protocol handler. The network management tier may also include modules facilitating the monitoring and enforcement of service level maintenance functions in support of Service Level Agreements (SLAs).

In embodiments of the invention, the network application server is implemented by use of a hardware configuration which may include one or more of the following: one or more networking processors dedicated to the forwarding engine, one or more general execution processors dedicated to the applications and network management tiers, two or more Ethernet ports, RAM, and a flash memory. Modules on the network application server are executed on the general execution processors. In some embodiments of the invention, the forwarding engine is encoded in microcode, while the modules are implemented in a platform independent, object-oriented language, such as JAVA ™. The separation between the processors supporting the forwarding engine and the application processors allow packets to be streamed through the forwarding engine continuously, irrespective of loading, unloading, modification, or failure of one or more modules running on the general execution processors.

In embodiments of the invention, the network application server may be configured to operate in parallel with similar devices. For instance, a cluster of network application servers may be stacked, in order to facilitate distributed processing and redundancy. In embodiments of the invention, stacked servers may be coupled by a local network or via a WAN, such as a service provider network or the Internet. In embodiments of the invention, the devices may be stacked, or coupled, by daisy chaining; in other embodiments, the devices may be coupled via a hub configuration. In embodiments of the invention, the modules are executed as threads distributed over multiple network application

servers. These and other aspects and embodiments of the invention shall be elaborated herein.

BRIEF DESCRIPTION OF THE FIGURES

Fig. 1 illustrates a network architecture with network application servers present at subscriber sites according to embodiments of the invention.

Fig. 2 illustrates a network architecture with network application servers located in a service provider network according to embodiments of the invention.

Fig. 3 illustrates a modular software architecture used in embodiments of the invention.

Fig. 4 illustrates a hardware implementation of the network application server in embodiments of the invention.

Fig. 5 is a flow chart for a boot up process for the network application server.

Fig. 6 illustrates a plurality of network appliances coupled via a local network according to an embodiment of the invention.

Fig. 7 illustrates a plurality of network servers distributed over the Internet.

Fig. 8 illustrates a plurality of network application servers coupled in a daisy chain according to an embodiment of the invention.

DETAILED DESCRIPTION

A. System Overview

The present invention comprises an architecture for programmable network application servers. The network server facilitates the inclusion of modules that support assorted applications and network functions. Modules executed on the network application server may perform operations on network traffic routed through the network server.

Modules supporting the applications and network functions may be loaded or unloaded dynamically, through one or more networks coupled to the network applications server. The modules may be loaded, modified, or deleted without restarting the network server or disrupting any network traffic routed through the network applications server. Modules residing on a server may also be reprogrammed via the network, without interruption of traffic passing through the server.

In embodiments of the invention, the network application server may be used to connect a network service provider, such as an ISP, to a LAN or terminal used by a subscriber of the ISP. In some such embodiments, the network application server may be physically present at the subscriber's site. One such embodiment is illustrated in Figure 1. A service provider network 100, or backbone, is coupled a subscriber LAN 102 106 through a network application server 104 108. Network traffic flows between the LAN 102 106 and the service provider network 100 via the network application serer 104 108, which may be physically present on the subscriber's site. Modules resident in

the network application server 104 108 may perform operations on the network traffic between the LAN 102 106 and the service provider network 100.

In alternative embodiments, the network application servers may reside within the service provider network. Such an embodiment is illustrated in Figure 2, in which the network application server 200 202 resides within the service provider network 100 and is coupled to the subscriber LAN 102 106, via a network connection which may employ a VPN 204 206.

The modular software architecture of the invention provides numerous advantages over prior art systems. The network application server may be configured, installed, and commissioned by the service provider without necessitating a truck-roll. In particular, the invention allows network service providers to administer the network application server remotely, via the service provider network 100. This eliminates the overhead of dispatching manpower to administer the server. The invention also decouples application and network management software from the hardware on the network server. This confers numerous advantages over prior art systems, allowing third party vendors to create modules for existing network servers; allowing existing network servers to support novel applications by loading modules supporting the respective applications; and allowing the service provider to sell the network server to subscribers independently from the hardware. The inclusion of new applications on existing network servers obviates the need to replace hardware on the servers to support new applications. Moreover, the network server's

ability to load, unload, and modify modules without disrupting traffic routed through the server is unprecedented.

The server may also be configured to operate in parallel with similar devices. For instance, a cluster of network appliances may be configured to facilitate distributed processing and redundancy. These appliances may be coupled via a local network connection, or over the service provider network.

This specification shall proceed by elaborating upon the following aspects of the invention:

- a modular software architecture employed by embodiments of the invention
- hardware implementations of the invention
- distributed processing amongst network application servers
- alternative embodiments of the invention

B. Modular Software Architecture of the Network Application Server

The programmable network application server is designed to support multiple network functions, which may be dynamically loaded, modified or deleted from the network application server 104 108 200 202 remotely via the service provider network 100. In embodiments of the invention, each network function supported by the server is performed by a software module dedicated to the network function. The software module may be loaded onto the network application server either prior to deployment or via the service provider network 100 at any time in its operation. Conversely, modules may be either updated

with patches to reflect updates to the corresponding network function, or deleted if the corresponding network function is no longer to be supported on the device.

The individual modules may be coded in an object-oriented language that is executed on a platform supported by processors in the network application server. For instance, in a non-limiting embodiment of the invention, the individual modules may be written in Java ™. Other languages suitable for the modules will be apparent those skilled in the art. The platform for one or more processors in the network application server may be an operating system specialized for network system such as VxWorks ™ 5.4. Alternative operating systems include various versions of Linux. Other alternatives will also be apparent to those skilled in the art.

A software architecture used to support the network functions of the present invention is illustrated in Figure 3. The multi-tier software architecture 300 includes a Management Interface Tier 302, an Applications Tier 304, and a Forwarding Engine 306. The Forwarding Engine may be coupled to the subscriber LAN 102 and to the service provider network 100. The Management Interface Tier 302 and Applications Tier 304 include modules for supporting network management functions and applications, respectively. The Forwarding Engine 306 is responsible for forwarding packets between the service provider network 100 and the subscriber LAN 102. The forwarding engine 306 also comprises the interface between the data packets routed through the network application server 104 108 200 202 and the modules resident in the management

interface tier 302 and the applications tier 304. The forwarding engine also includes encryption/decryption facilities for allowing administrative access to the modules resident in the management interface layer 302 and applications layer 304. Alternatively, encryption facilities, such as DES, may be supported by a Digital Signal Processor (DSP) 336. Signal features of the software architecture are described herein.

Forwarding Engine

In embodiments of the invention, the forwarding engine 306 is responsible for one or more of the following

- forwarding packets between the subscriber LAN 102 106 and the service provider network 100, and
- security services for the network application server 104 108 200 202

In embodiments of the invention, the forwarding engine 306 is coupled to the subscriber's LAN 102 and the service provider network 100 in order to facilitate the transport of data packets between the LAN 102 and the service provider network 100. In some such embodiments, the forwarding engine 306 also serves as the conduit between the data packets routed through the network applications sever and the modules resident in the management interface tier 302 and applications tier 304. The forwarding engine 306 may also be coupled to other network application servers, as described further in section E of this specification.

In embodiments of the invention, the forwarding engine supports a minimum line rate of 1 GB/sec or greater. In embodiments of the invention, the

forwarding engine is also resident on one or more dedicated networking processors, in order to ensure that packets continue to be routed by the forwarding engine during the loading, unloading, modification, or execution of modules in the management interface tier or application tier of the network application server. Hardware implementations supporting these features of the forwarding engine are elaborated further in section C of this specification.

The forwarding engine may also include security features for the network application server. As described earlier, a network service provider administering the network application server 104 106 200 202 may access the modules resident in the network access server. In order to restrict access to the modules to the particular network service provider, the forwarding engine may include an authentication system. In some embodiments of the invention, the authentication system may include a system such as Kerberos. The forwarding engine may also include an encryption system such as DES. Hardware implementations discussing these security features are described in section C of this application.

Application Tier

The Application Environment Tier 304 may include one or more loadable modules 324-334. The applications which may be supported by the modules may include VPN applications such as Multiprotocol Label Switching, or MPLS, Layer Two Tunneling Protocol, or L2TP, IP Sec. This allows the network application server to emulate any type of VPN server. The modules may also be unrelated to VPNs, and support applications such as QoS or

Multicasting. As an example, modules may support caching at the network appliance, to store data such as multimedia files. Modules in the application tier may also be encoded to support entirely new types of applications. Data flows which are to be processed by application modules are forwarded to the respective applications modules by the forwarding engine 306.

Management Interface Tier

In embodiments of the invention, the Management Interface Tier 302 may include a plurality of individual modules 312 – 322 which support network management functions. These functions may include, but are not limited to :

- remote network monitoring and management protocols, such as the Simple Network Management Protocol (SNMP) and the Common Management Information Protocol (CMIP)
- Messaging protocols based upon CORBA and XML
- monitoring and enforcement of service level maintenance functions in support of Service Level Agreements (SLAs)

Some advantages of the present invention are evidenced by the type of modules supported by the Management Interface Tier 302. Illustrative examples of such modules include the SLA and caching modules

Service Level Agreements (SLA)

One or more of the modules in the Management Interface Tier 302 may support novel SLA functions. In particular, such modules allow service

providers or network administrators to be alerted to network parameters in real-time. As an illustrative, non-limiting example, a Service Level Agreement module in the network appliance may monitor the network for traffic discards. When traffic discards increase beyond a pre-determined threshold, the SLA module may send an e-mail message directly to an administrator of the network which indicates the nature of the problem. Alternatively, upon detecting a high number of interface resets, the SLA module may send an alarm message to an administrator, immediately message a stand-by network application server to begin processing all traffic, and then await repair.

In embodiments of the invention, the administrator message may contain a link to a URL for the appliance. Upon clicking on the URL for the appliance, the administrator may receive an image indicating relevant network performance metrics. This in turn allows the administrator—as an example—to contact a network service provider regarding the failure. In alternative embodiments, the message may be forwarded directly to the service provider.

Messaging Standards

Modules in the Management Interface Tier 302 may support a range of open or proprietary messaging standards. In embodiments of the invention, the messaging standards may be based on CORBA or XML. For instance, a module may support a CORBA Object Request Broker (ORB), for communication between objects on the LAN, or between objects on the LAN and the Internet. Another module in the Management Interface Tier 302 may

support the XML-based Simple Object Access Protocol (SOAP). Such a module may store and interpret DTDs at the network device

The Management Interface Tier allows service providers to upload proprietary modules which allow the service provider to monitor network parameters remotely and in real-time. These modules may be developed internally by the service provider. They also may be installed remotely on the network application server 104 108 200 202 via the service provider backbone 100 or may be loaded on the network application server 104 108 200 202 prior to installation.

C. Hardware Implementations of the Network Application Server

A hardware implementation used in embodiments of the invention is illustrated in Figure 4. The network application server 400 includes a first communications port 402 and a second communications port 404. In some embodiments. one or more of the ports 402 404 are Gigabit Ethernet ports and/or 10/100 Ethernet ports. The ports 402 404 are coupled to a processor cluster 406. The processor cluster 406 includes the forwarding engine 306, which resides on one or more dedicated networking processors; an application engine 410 comprising one or more processors dedicated to the application tier and network management level modules; one or more flash memory chips 412; and random-access memory (RAM) 414; all coupled by a local bus 416.

The processors in the forwarding engine 306 may include one or more processors dedicated to encryption/decryption functions 336 and authentication protocols. In embodiments of the invention, DES is used as an encryption

standard. The forwarding engine serves as a gateway between the ports 402 404 and the remaining processors in the network application server, including the application engine 410, flash memory 412, and RAM 414. The processors in the application engine 410 are dedicated to the modules in the application tier 304 and network management layer of the software architecture. In other embodiments, application and network management modules run on separate, dedicated processors. The flash memory 412 is used to store microcode, which is super-saved for error recovery. By the inclusion of multiple dedicated processors 306 410 412 414, the processor cluster 406 effectively decouples network functions from applications. In embodiments of the invention, the processors in the network application server support line rates of 1 GB/sec or better.

As explicated in Section B, the forwarding engine 306 is also responsible for the transmission of data packets between the service provider network and the subscriber's LAN. Placing the forwarding engine on dedicated processors enables the network application server to be more resilient to failure during the loading, unloading, and execution of modules. In particular, packets continue to be forwarded by the forwarding engine irrespective of activity on the processors comprising the application engine. For instance, should a module crash on the application engine, it is possible to establish rules which cause the forwarding engine to continue to forward packets, allowing the service provider to access the forwarding engine to perform repairs.

D. Initialization of the Network Application Server

The network application server of the present invention is designed to be operational immediately, without requiring manpower at the network application server on the part of the service provider. In particular, the network application server may be shipped to a subscriber and installed at the subscriber site by the subscriber's personnel. This facility may be accomplished by different embodiments of the invention.

An initialization procedure for the network application server is illustrated in the flow chart of Figure 5. Once the network application server is powered up for the first time 500, processors in the forwarding engine load microcode into local DRAM 502, which can then be run by the processors in the forwarding engine. The microcode may be retrieved either from flash memory or from the service provider network. The forwarding engine then goes to a server on the service provider network 504 to retrieve modules and configuration data for the application engine. These modules and configuration data are then loaded onto one or more processors in the application engine.

E. Stackable Architecture

In embodiments of the invention, multiple network application servers may be stacked in order to 1) balance loads amongst the network devices and 2) provide redundancy amongst modules. A plurality of stacked network devices is illustrated in Figure 6. The stack 600 includes a plurality of network devices 604-614, coupled through network interfaces 616-626. In embodiments of the invention, the network interfaces 616-626 are Gigabit Ethernet interfaces. In

the embodiment illustrated in Figure 6, each network device 604-614 includes one or more modules 324-334 from the Application Tier 304 and one or more modules 312-322 from the Management Interface Tier 302. The stacked arrangement allows packets arriving from the service provider network 100 to be processed by the network devices 604-614 in parallel. Additionally, stacking network devices 604-614 enables redundancy in case of failure of one or more of the modules 312-334.

In other embodiments, stacks of network devices may be coupled via a service provider network, or other internetwork. Such an embodiment is illustrated in Figure 7. A first plurality of stacked network devices 700 are locally coupled via network interfaces 716 718 720. Each of the individual devices 704 706 708 within the first stack 700 includes a module from the network management interface 312 314 316 and a module from the application layer 324 326 328. A second plurality of stacked network devices 702 is coupled to the first plurality 700 via the service provider network 100. Each of the devices 722 724 726 in the second stack also includes a module from the network management interface 318 320 322 and a module from the application layer 330 332 334.

The two local stacks of network devices 700 702 comprise a single distributed stack which operates as though the individual network devices 704 706 708 710 712 714 are in a single locally coupled stack. As such, the distributed stack illustrated in Figure 7 supports load balancing and redundancy

amongst the individual modules, as described above for the local stack of Figure 6.

Daisy Chaining

In some embodiments, network application servers may be coupled by daisy-chaining. One such embodiment is illustrated in Figure 8. The figure depicts three network application servers 800 802 804. Each network application server 800 802 804 includes a first port 806 808 810 and a second port 812 814 816. The network servers are coupled to each other, the subscriber LAN 818 and the service provider network 820 as follows: the first port 806 in the first network application server 800 is coupled to the subscriber LAN 102. The second port 816 in the first network server 800 is coupled to the first port 808 in the second network server 802. The second port 814 in the second network server 802 is coupled to the first port 810 in the third network server 804. The second port 812 in the third network server 804 is coupled to the service provider network 100. As will be apparent to one skilled in the art, any number of network application servers can be coupled through daisy chaining.

In some embodiments of the invention, stacked network application servers may contain duplicate copies of modules, in order to facilitate redundancy. In embodiments of the network application server, the processors in the processor cluster 406 may include a distributed operating system, which executes the modules in threads distributed over multiple stacked network application servers. Other embodiments facilitating distributed processing will be apparent to those skilled in the art.

F. Alternative Embodiments

In alternative embodiments, network server may reside in a device such as a router or switch. The network server may also comprise a self-contained unit which supports routing protocols and algorithms, such as OSPF. Other equivalent embodiments will be apparent to those skilled in the art.

G. Conclusion

The foregoing description of various embodiments of the invention has been presented for purposes of illustration and description. It is not intended to limit the invention to the precise forms disclosed. Many modifications and equivalent arrangements will be apparent.